

## **AMENDMENTS TO THE SPECIFICATION:**

Please amend paragraphs 0029-0030 of the Specification as follows:

**[0029]** The GPPI 116 is a generic framework that abstracts services of the cellular modem software ~~408~~ 110 for access by the generic TSY 114. The generic TSY 114 may be used with any type of hardware and/or cellular modem software that includes the appropriate GPPI 116. This allows the TSY 114 to be easily ported and/or reused with various hardware and software architectures.

**[0030]** In one embodiment, the GPPI 116 operates by abstracting the interactions between the generic TSY 114 and cellular modem software ~~408~~ 110 into predetermined categories, or primitives. Each operation, therefore, between the generic TSY 114 and cellular modem software ~~408~~ 110 may be formed by a combination of one or more of these primitives. The GPPI 106 acts as an intermediary that translates the generic primitives used by the TSY 114 into concrete primitives used to form commands of the cellular modem software ~~408~~ 110.

Please amend paragraphs 0034-0035 of the Specification as follows:

**[0034]** Generally, the abstract primitive factory 312 is based on the abstract factory design pattern described in “Design Patterns” by Gamma, et al. The abstract primitive factory 312 ~~240~~ provides an interface for creating families of related or dependent objects (e.g., “call request”, “call confirm”) without specifying implementation specific, or concrete, information. This allows the users of the abstract primitive factory 312 (in this case the TSY 304) to perform cellular modem software-type operations using primitives without requiring any knowledge of the underlying cellular modem software 320.

**[0035]** The abstract primitive factory 312 is an interface used to create the concrete primitives 310. It is up to the concrete factory ~~316~~ 314 to actually create the concrete primitives 310. The concrete primitive factory 314 contains the logic that determines how the abstract primitives 308 are translated into concrete primitives 310. There may be various mappings (e.g., one-to-one, one-to-many) of abstract primitives 308 to concrete primitives 310, as well as different data structures used by the primitives.